

The ATLAS Simulation Software

Z. Marshall^{a,b} for the ATLAS Collaboration

^aCalifornia Institute of Technology, Physics Department, Pasadena, CA 91125, United States of America

^bColumbia University, Nevis Laboratory, 136 So. Broadway, Irvington, NY 10533, United States of America

We present the status of the ATLAS Simulation Project. Recent detector description improvements have focussed on commissioning layouts, implementation of inert material, and comparisons to the as-built detector. Core Simulation is reviewed with a focus on parameter optimizations, physics list choices, visualization, large-scale production, and validation. A fast simulation is also briefly described, and its performance is evaluated with respect to the full Simulation. Digitization, the last step of the Monte Carlo chain, is described, including developments in pile up and data overlay.

1. Introduction

ATLAS [1], one of the general-purpose detectors at the Large Hadron Collider [2], began operation in 2008. The detector will collect data from 14 TeV proton-proton collisions as well as 5.5 TeV per nucleon pair heavy ion (Pb-Pb) collisions. During proton-proton collisions at design luminosity of $10^{34} \text{ cm}^{-2}\text{s}^{-1}$, beam bunches will cross every 25 ns and provide on average 20 collisions per bunch crossing. ATLAS has been designed to record 200 bunch crossings per second, keeping only the most interesting interactions for physics searches.

In order to study detector response and the effectiveness of proposed search strategies, a detailed simulation has been implemented that carries events from Monte Carlo generation through to output in a format identical to that of the detector. The software is integrated into the Athena software framework [3] and uses PYTHON as a scripting language for the runtime configuration of jobs. Libraries are loaded on demand, keeping each job as light as possible in memory.

The Monte Carlo data chain is generally divided into three steps, though they may be combined into a single job: generation of the Monte Carlo event and immediate decays, simulation of the detector and physics interactions within it, and digitization of the energy deposited in the

sensitive regions of the detector into voltages for comparison to the readout of the real ATLAS detector. The output of the simulation chain can be presented in either object format or in a format identical to the output of the ATLAS data acquisition system. Thus, both the Monte Carlo and real data from the detector can be run through the same ATLAS trigger and reconstruction packages. The data flow of the Monte Carlo chain can be seen in Figure 1. Algorithms to be run are represented by square-cornered boxes, and persistent data objects are placed in round-cornered boxes. The optional steps required for pile up or event overlay are shown with a dashed outline.

The Monte Carlo chain, divided in this way, uses resources more effectively and simplifies software validation. Event generation jobs, typically quick and with small output files, can be run for several thousands of events at a time. By storing the output rather than regenerating it each time, it becomes possible to run identical events through different versions of the simulation software or with different detector configurations. The simulation step is particularly slow, and can take several minutes per event. Simulation jobs are therefore divided into 25-50 events at a time; only a few events may be completed in a single heavy ion collision simulation job. Digitization jobs are generally configured to run ~ 1000

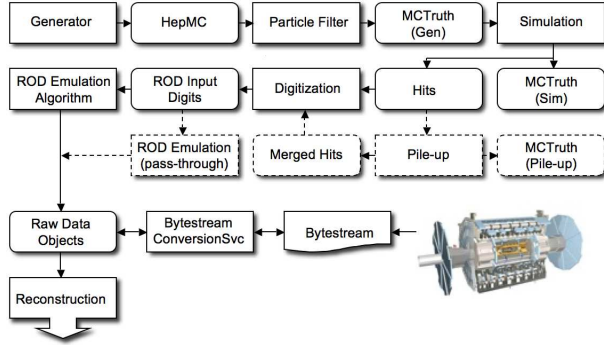


Figure 1. The general layout of the ATLAS simulation. The flow can be seen from job option input to Monte Carlo generators through Monte Carlo identical to the data read out from the detector.

events in order to produce 2 GB files and thereby ease file handling. As much as is possible during this chain, options that were set in a previous step are used to automatically configure the subsequent step. In that way, for example, the detector geometry used for a simulation job and subsequent digitization job are consistent.

Large-scale production is done on the LHC Computing Grid (Grid) [4]. A single task on the Grid is separated into many jobs depending on the content and complexity of the task. A job can be completed by a single CPU within the maximum allowed time for a job on the Grid (typically 2-3 days). In the case of a full chain of jobs being run (generation, simulation, and digitization), each subsequent step is automatically held in the queue until the required data is available from the previous step.

2. Detector Layouts

The ATLAS simulation, digitization, and reconstruction each run in distinct jobs, but they must be able to use the same detector geometry. It is desirable, therefore, to maintain a complete geometry description that can be used by each step and is not specific to any. ATLAS uses GeoModel [5], a library of geometrical primitives,

to describe and construct the detector. For the digitization and reconstruction, this detector description is entirely sufficient to place hits, reconstruct tracks and objects, and complete all necessary calculations. The number of physical volumes and memory required by subdetector for the full ATLAS detector layout is shown in Table 1.

Subsystem	Volumes	Memory
Inner Detector	56,838	13,798
Calorimetry	182,262	43,448
Muon System	76,945	30,704
ATLAS TOTAL	316,043	87,950

Table 1

Numbers of physical volumes and memory in kilobytes required to build various pieces of the ATLAS detector in GeoModel.

The ATLAS detector geometry used for simulation, digitization, and reconstruction is built from databases containing both the primary numbers describing the physical construction and conditions data. The latter contains all the necessary information to emulate a single data-taking run of the real detector. Subdetectors may be individually misaligned. By using the geometry databases, it is already possible for the different steps of the Monte Carlo chain to read identical detector constants and run conditions.

The same geometry and simulation infrastructure supports all the test stands and installation configurations of the ATLAS detector. Layouts are supported for combined and standalone test beams, cosmic ray data taking stands, and the ATLAS detector during installation. The latter include shifted endcaps for inner detector installation, for example.

The GeoModel descriptions of most ATLAS subsystems are built using constants in the geometry database. Recently, a package has been constructed that parses an XML description of a detector's geometry and builds a transient representation from GeoModel primitives at runtime. This generic package can translate any valid XML description of detector geometry into GeoModel

format, although thus far it has been used only for describing geometry of the muon systems rather complicated dead material.

3. Simulation and Optimization

The ATLAS simulation uses GEANT4 8.3.patch02 [6,7]. In order to provide PYTHON flexibility to the GEANT4 simulation, an additional layer of infrastructure is necessary. “Standard” GEANT4 simulation typically runs from compiled C++, and in order to modify any of the parameters or the geometry used in the simulation it is necessary to recompile. In the Framework for ATLAS Detector Simulation (FADS), wrappers of several GEANT4 classes are implemented in order to allow selection and configuration without recompilation of any libraries.

The QGSP_BERT physics list, recently adopted for large-scale production, includes the Bertini intranuclear cascade model which was found to better describe hadronic shower shapes observed in test beam stands [10]. The list also includes step-limiting multiple Coulomb scattering which improves agreement in cosmic ray data with inner detector observables and reduces the effect of non-physical simulation parameters like GEANT4 range cuts. Neutrinos are not simulated, nor are neutrons after 150 ns. The latter has little impact on calorimeter resolution or response but saves both computing time and disk space.

Modifications to the transportation of charged particles through a magnetic field are underway. Infrastructure exists for separately controlling the parameters for propagation by particle type, energy, and region of the detector. In the calorimetry, for example, low-energy electrons need not be tracked as carefully as high-energy muons.

4. Digitization

The digitization step of the Monte Carlo chain takes GEANT4 hits and converts them into output identical to that of the detector (e.g. voltage and time). During this step, detector noise and additional background events are overlaid (“pile up”).

To completely simulate pile up as it will occur during data taking, several kinds of events must be overlaid on top of hard scattering signal events. Multiple minimum bias events are overlaid, with a configurable mean number of interactions. Minimum bias events in relevant bunch crossings before and after the signal event are also overlaid. A luminosity-dependent gas of neutrons and photons provides a constant source of background in the muon system (“cavern background”). Interactions with gas in the imperfect vacuum of the beam pipe produce beam gas events, and interactions of protons with upstream collimators produce beam halo events. All of these types of events are overlaid at a configurable rate, sometimes including a safety factor to account for uncertainties in the real rate.

Infrastructure for overlaying data from the real detector is under development. Eventually, all of the above types of backgrounds will be taken directly from data.

5. Fast Simulation

Almost 80% of the full simulation time is spent simulating particles traversing the calorimetry, and about 75% of the full simulation time is spent simulating electromagnetic particles. The Fast G4 Simulation aims to speed up the slowest parts of the full simulation [8]. The approach taken, therefore, is to remove low energy electromagnetic particles from the calorimeter and replace them with pre-simulated showers loaded into memory during initialization. Using this approach, computing time is reduced by a factor of three in hard scattering events with little physics penalty. This simulation may eventually become the default simulation for all processes that do not require extremely accurate modeling of calorimeter response or electromagnetic physics.

6. Validation

Several types of automatic tests have been developed to ease validation of the software chain. Basic functionality tests are run on all of the ~ 50 copies of the ATLAS software that are compiled every night. A small set of releases run somewhat

more complicated tests daily, including single particle simulation and digitization of a small number of events. An additional set of tests exercises the entire Monte Carlo chain from end to end, using the previous day's output of a given step as the input for the subsequent step on the following day. Each set of tests monitors functionality, output, and computing resource consumption. The output of each release, after distribution to the production system, is compared in detail with the output of previously validated releases. As much as is possible, an effort is made to compare output to whatever data is available, including test beams and cosmic ray data.

Current simulation takes 2-15 minutes per event on most modern computers, depending upon the complexity and activity of the event. Table 2 lists fast and full simulation times per event, in kSI2K seconds, for various types of events. Over 100 million events have been simulated in 2008 with GEANT4 in large-scale production. The crash rate, excluding problems with the production system, is below 10^{-6} .

Sample	Full Sim.	Fast Sim.
H(130) to 4ℓ	2300	840
Minimum Bias	960	410
SU3 SUSY	3400	1300
$Z \rightarrow \mu^+ \mu^-$	1800	780
$Z \rightarrow e^+ e^-$	2200	720
$Z \rightarrow \tau^+ \tau^-$	1800	720

Table 2

Simulation times per event, in kSI2K seconds, for the fast and full simulations. All times are averaged over 200 events.

7. Summary

ATLAS has achieved a stable, configurable detector simulation. The entire Monte Carlo chain is PYTHON-configurable, so that there is no need for users to recompile scripts in order to modify jobs. The same framework is used for test stands, cosmics, and the as-built detector.

Realistic detector conditions and misalignments are included. Users will soon be able to apply conditions to mimic a single data-taking run. Detector backgrounds are being studied, and data will eventually be overlaid on Monte Carlo to model these backgrounds. Thus far, tests of physics agreement are based mostly on the 2004 combined test beam.

Large-scale production is ongoing, with over 100 million events simulated so far in 2008. The simulation crash rate is below 10^{-6} .

REFERENCES

1. The ATLAS Collaboration, G. Aad et al., The ATLAS Experiment at the CERN Large Hadron Collider, JINST 3 (2008) S08003.
2. L. Evans, The Large Hadron Collider, New J. Phys. 9 (2007) 335.
3. Athena Core Software, see <http://atlas-computing.web.cern.ch/atlas-computing/packages/athenaCore/athenaCore.php>
4. I. Bird (ed.) et al., LHC computing Grid. Technical design report, CERN-LHCC-2005-024 (2005).
5. J. Boudreau and V. Tsulaia, The GeoModel Toolkit for Detector Description, Computing in High Energy and Nuclear Physics 2004 Conference (CHEP2004), Interlaken, Switzerland, September 27-October 1, 2004 (2004).
6. S. Agostinelli et al., Geant4 - a simulation toolkit, Nucl. Instr. Methods Phys. Res. A 506 (2003) 250-303.
7. J. Allison et al., Geant4 Developments and Applications, IEEE Transactions on Nuclear Science 53 (2006) 270-278.
8. E. Barberio et al., The Geant4-Based ATLAS Fast Electromagnetic Shower Simulation, ATL-SOFT-CONF-2007-002 (2007).
9. A. Rimoldi et al., First Report of the Simulation Optimization Group, ATL-SOFT-PUB-2008-002 (2008).
10. T. Carli, H. Hakobyan, A. Henriques-Correia, and M. Simonyan, Measurement of Pion and Proton Longitudinal Shower Profiles up to 20 Nuclear Interaction Lengths with the ATLAS Tile Calorimeter, ATL-TILECAL-PUB-2007-008 (2007) and notes in preparation.